

(E)ER, UML & RELATIONAL MODEL

ER Modeli:

- Entity (varlık) => Table (UML'de object, entity yerine geçer)
- Relationship (bağıntı)
- Attribute (nitelik) => Column

Varlık kümesi: benzer varlıkların oluşturduğu küme

Bağıntı kümesi: aynı türdeki benzer bağıntıların oluşturduğu küme

Domain: bir niteliğin tipi, değer aralığı, formatı

Bağıntının derecesi: bir bağıntının uçlarındaki varlık sayısı

Cardinality ratio (eleman sayıları): 1-1, 1-N, N-1 veya N-N olabilir

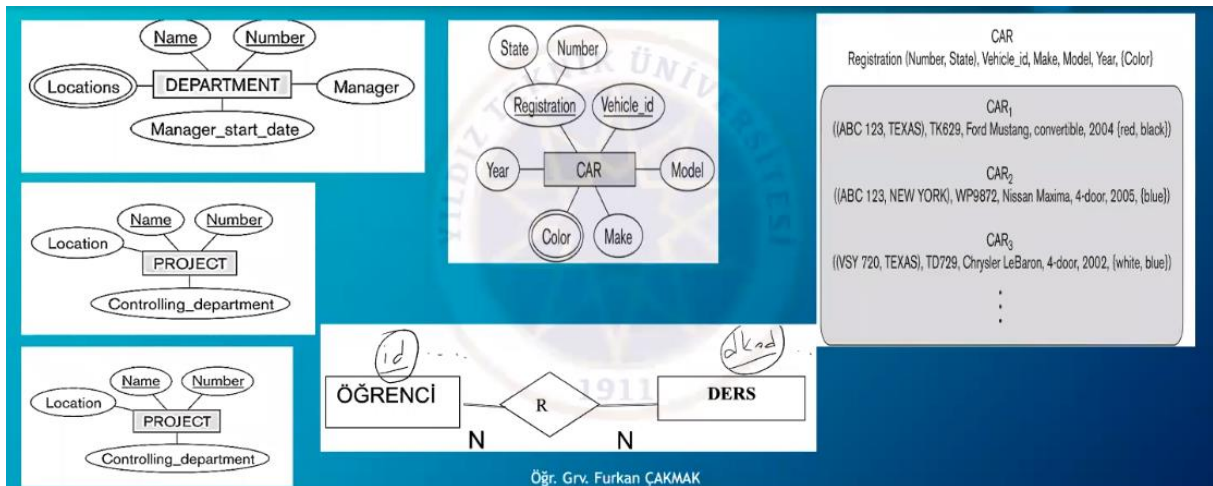
Candidate key: tek başına primary key olabilecek unique bir keydir (örn: TCNO)

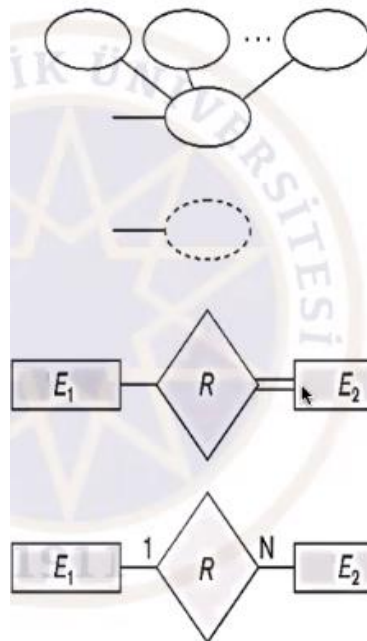
Primary key: candidate keyler arasından seçilmiş olan keydir

Super key: bir key ile beraber başka attributelerin kombinasyonu

Zayıf varlık: primary keyi olmayan varlık

Güçlü varlık: primary keyi olan varlık





Composite Attribute

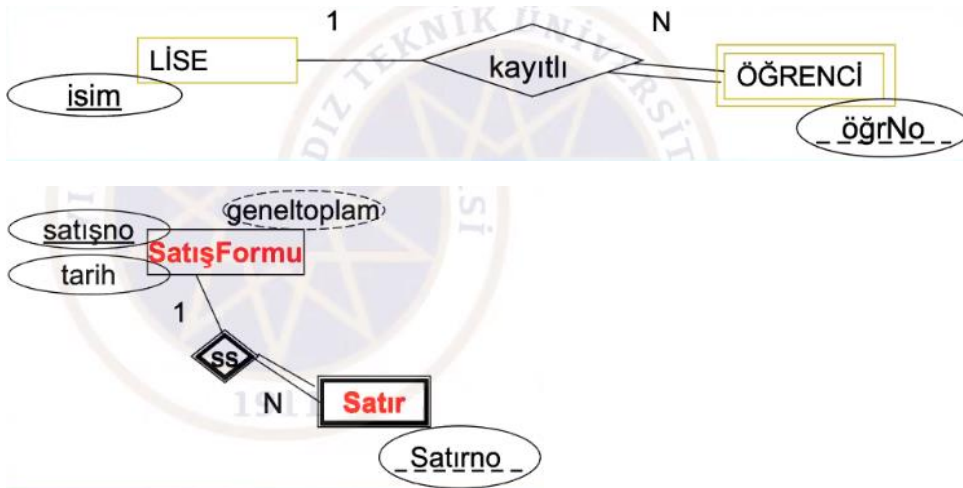
Derived Attribute

Total Participation of E_2 in R

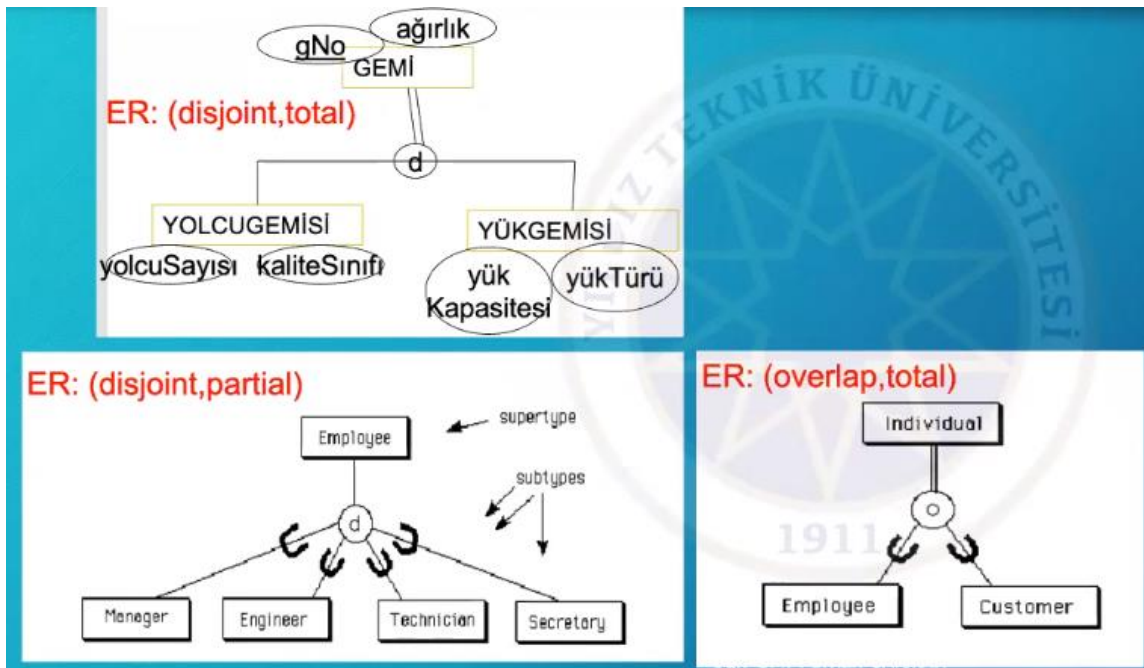
Cardinality Ratio 1: N for $E_1:E_2$ in R

Concept	Representation & Example
Degree	
recursive	
binary	
ternary	
Connectivity	
one-to-one	
one-to-many	
many-to-many	
Existence	
optional	
mandatory	

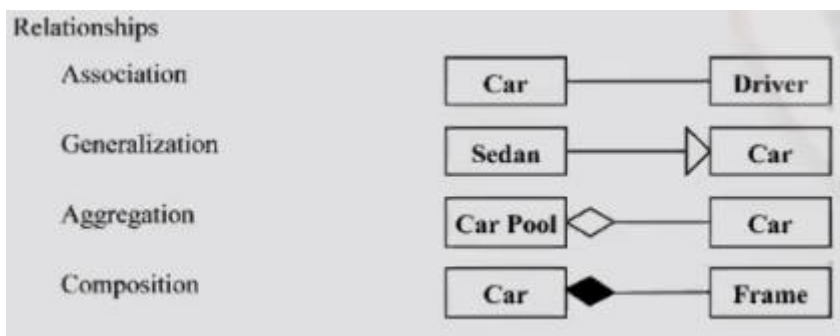
Zayıf Varlık Kümesi Örnekleri:



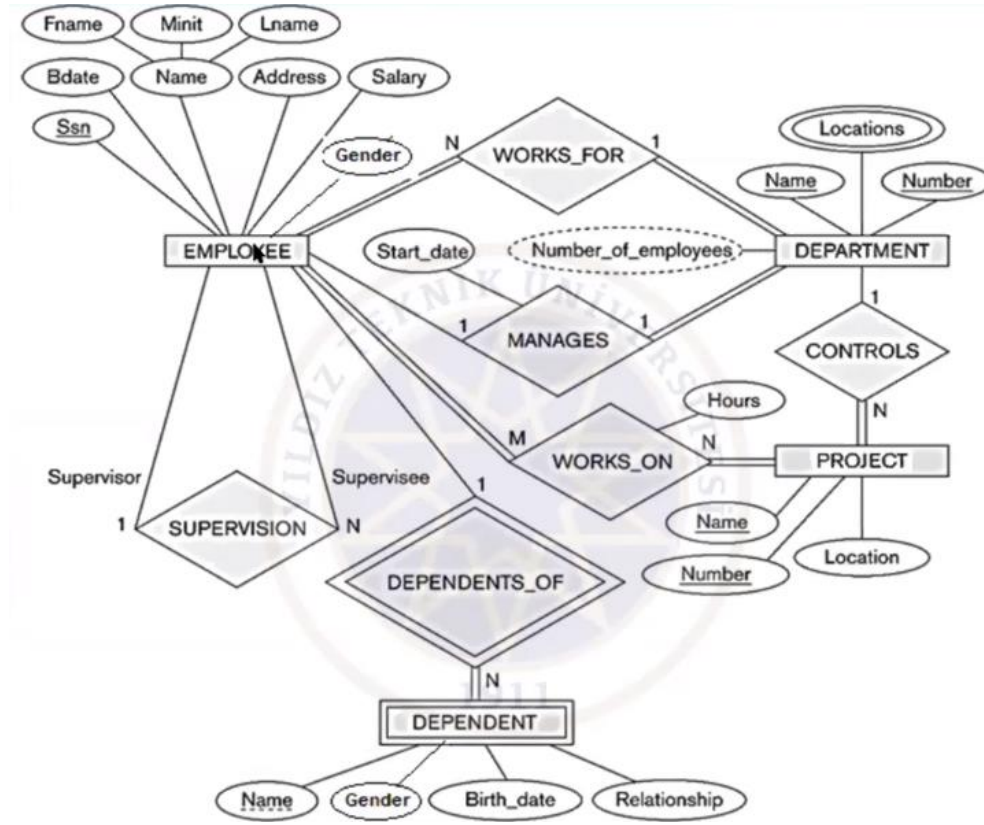
EER-Inheritance:



UML Gösterimi:



Örnek ER Tasarımı:



Informal Terms (Pratik model)	Formal Terms (Biçimsel model)
Table	Relation
Column Header	Attribute
All possible Column Values	Domain
Row	Tuple
Table Definition	Schema of a Relation
Populated Table	State of the Relation

Key constraint: tek primary key vardır, her key super keydir, her super key key değildir

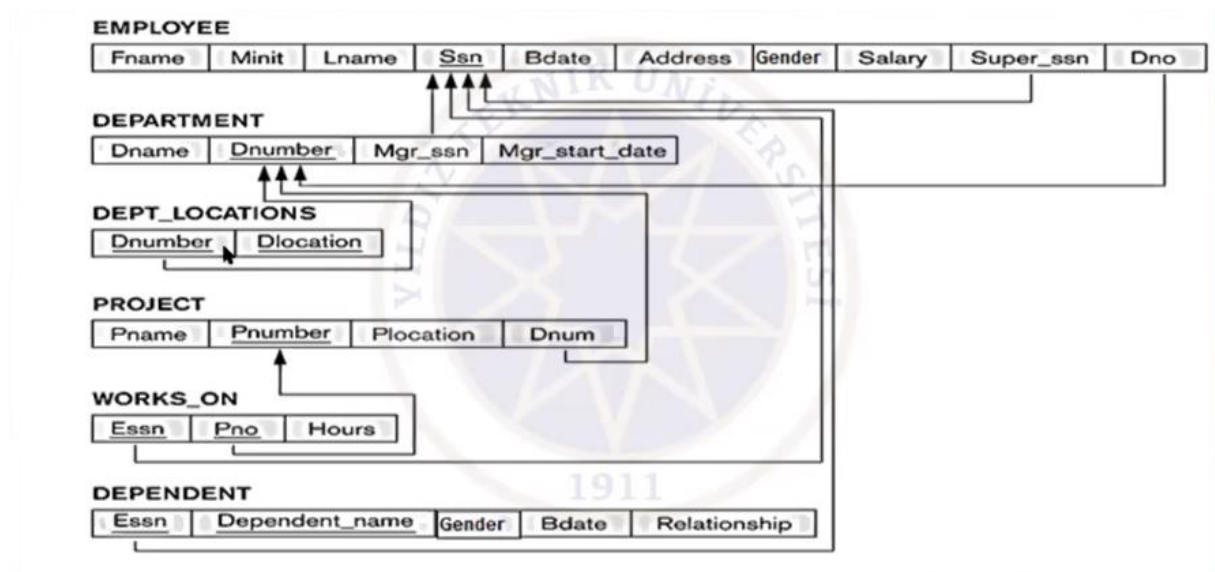
Entity integrity: pk null olamaz ve unique olmalıdır gibi bütünlük kısıtları

Referential integrity: fk ya null'dır ya da başka tabloda bir yeri referans göstermelidir

Domain constraint: yaş 0-100 arasıysa 105 giremezsin gibi

Semantic constraint: %5'ten fazla zam yapamazsın gibi (trigger ile manuel tanımlanmış)

Örnek Relational Schema Diagram:



ER vs Relational Model:

ER Model	Relational Model
Entity type	"Entity" relation
1:1 or 1:N relationship type	Foreign key (or "relationship" relation)
M:N relationship type	"Relationship" relation and two foreign keys
n -ary relationship type	"Relationship" relation and n foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

RELATIONAL ALGEBRA

<ul style="list-style-type: none"> • SELECT: σ • Project: Π • Sort: S • Rename: ρ • Extend: E • Aggregate: \mathcal{F} • Groupby: \mathcal{F} 	<ul style="list-style-type: none"> • Union: \cup • Intersection: \cap • Set Difference: $-$ • Division: \div • Product: \times • Theta-join: \bowtie • Natural-join: $*$ veya \bowtie • Semijoin: \ltimes • Antijoin: \rhd • Full Outer Join: \bowtie • Left Outer Join: \ltimes • Right Outer Join: \bowtie
---	---

Select:

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

$$\sigma_{A=B \text{ ve } D > 5}(r)$$

A	B	C	D
α	α	1	7
β	β	23	10

Project:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$$\Pi_{A,C}(r)$$

A	C
α	1
β	1
β	2

Union:

A	B
α	1
α	2
β	1

$$r \cup s$$

A	B
α	1
α	2
β	1
β	3

Intersect:

A	B
α	1
α	2
β	1

$$r \cap s$$

A	B
α	2

Difference:

A	B
α	1
α	2
β	1

 $r - s$

A	B
α	2
β	3

A	B
α	1
β	1

Rename:

Paternity		$\rho_{\text{Father} \rightarrow \text{Parent}}(\text{Paternity}) \cup \rho_{\text{Mother} \rightarrow \text{Parent}}(\text{Maternity})$	
Father	Child	Parent	Child
Adam	Cain	Adam	Cain
Adam	Abel	Adam	Abel
Abraham	Isaac	Abraham	Isaac
Abraham	Ishmael	Abraham	Ishmael
		Eve	Cain
		Eve	Seth
		Sarah	Isaac
		Hagar	Ishmael

Cartesian Product:

A	B
α	1
β	2

 $r \times s$

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Theta-Join:

Employee	Project
Smith	A
Black	A
Black	B

Code	Name
A	Venus
B	Mars

Employee	Project	Code	Name
Smith	A	A	Venus
Black	A	A	Venus
Black	B	B	Mars

 $\text{Employees} \bowtie_{\text{Project}=\text{Code}} \text{Projects}$

Natural Join:

Employee	Department
Smith	sales
Black	production
White	production

r_1

Department	Head
production	Mori
sales	Brown

r_2

Employee	Department	Head
Smith	sales	Brown
Black	production	Mori
White	production	Mori

$r_1 * r_2$ veya $r_1 \bowtie r_2$

Left, Right, Full Outer Join:

Employee	Department
Smith	sales
Black	production
White	production

r_1

Department	Head
production	Mori
purchasing	Brown

r_2

Employee	Department	Head
Smith	Sales	NULL
Black	production	Mori
White	production	Mori

$r_1 \bowtie r_2$

Employee	Department	Head
Black	production	Mori
White	production	Mori
NULL	purchasing	Brown

$r_1 \bowtie r_2$

Employee	Department	Head
Smith	Sales	NULL
Black	production	Mori
White	production	Mori
NULL	purchasing	Brown

$r_1 \bowtie r_2$

Semi Join:

Name	EmpId	Dept Name
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Production

Employee

Dept Name	Manager
Sales	Bob
Sales	Thomas
Production	Katie
Production	Mark

Department

Name	EmpId	DeptName
Sally	2241	Sales
Harriet	2202	Production

Employee \bowtie Department

Öğr. Grv. Furkan ÇAKMAK

Anti Join:

Name	EmpId	Dept Name
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Production

Employee

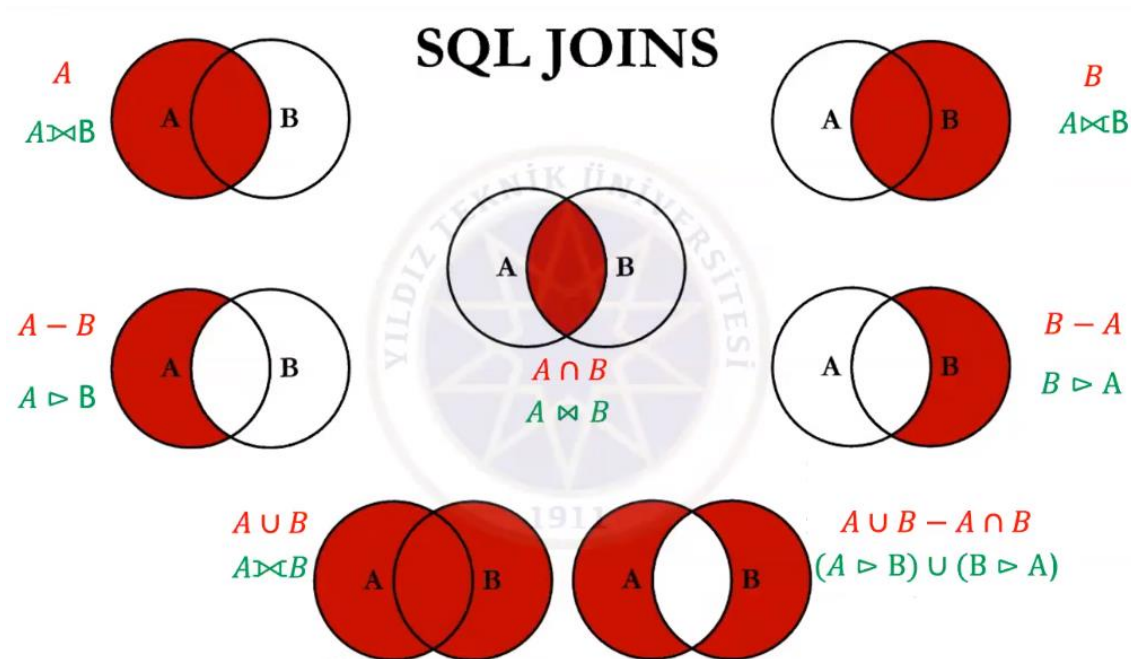
Dept Name	Manager
Sales	Sally
Production	Harriet

Department

Name	EmpId	DeptName
Harry	3415	Finance
George	3401	Finance

Employee ▷ Department

Joins Özet:



Division: ($R \div S = T$)

SSN_PNOS		SMITH_PNOS		R		S	
Essn	Pno	Pno		A	B	A	
123456789	1	1		a1	b1	a1	
123456789	2	2		a2	b1	a2	
666884444	3			a3	b1	a3	
453453453	1			a4	b1		
453453453	2			a1	b2		
333445555	2			a3	b2		
333445555	3			a2	b3		
333445555	10			a3	b3		
333445555	20			a4	b3		
999887777	30			a1	b4		
999887777	10			a2	b4		
987987987	10			a3	b4		
987987987	30						
987654321	30						
987654321	20						
888665555	20						

SSNS
Ssn
123456789
453453453

T
B
b1
b4

Aggregate:

- $\mathcal{F}_{MAX\ Salary}(Employee)$
- $\mathcal{F}_{MIN\ Salary}(Employee)$
- $\mathcal{F}_{AVG\ Salary}(Employee)$
- $\mathcal{F}_{COUNT\ SSN}(Employee)$
- $\mathcal{F}_{COUNT\ SSN,AVG\ Salary}(Employee)$

Groupby (with Aggregate):

$DNO \mathcal{F}_{COUNT\ SSN,AVG\ Salary}(Employee)$

Dno	Count_ssn	Average_salary
5	4	33250
4	3	31000
1	1	55000

Count_ssn	Average_salary
8	35125

Sort, Rename, Extend:

- Öğrencileri önce mezuniyet yıllarına sonra da isimlerine göre sırala
- Q7: $\mathcal{S}_{GradYear, SName}(STUDENT)$
- Bir önceki sorudaki SName alanını CSMajors olarak değiştirelim
- Q8: $\rho_{SName, CSMajors}(Q7)$
- STUDENT tablosuna, ilk mezun olan öğrenciden (1863) buyana kaç yıl geçtiğini hesaplayarak oluşturulan GradClass kolonunda ekleyiniz.
- Q9: $\mathcal{E}_{GradYear-1863, GradClass}(STUDENT)$
- Student tablosuna «College» kolonunu ekleyip içini «YTU» olarak doldurunuz.
- Q10: $\mathcal{E}_{YTU, College}(STUDENT)$

STUDENT(SId, SName, GradYear, MajorId)
DEPT(DId, DName)
COURSE(CId, Title, DeptId)
SECTION(SectId, CourseId, Prof, YearOffered)
ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

Groupby Örnekler:

- Herhangi bir öğrencinin minimum mezuniyet yılını bulunuz
- Q14: $\mathcal{F}_{MIN(GradYear)}(STUDENT)$
- Tüm öğrencilerin bölüm numaralarını tekrarlardan kurtularak bulunuz
- Q15: $\mathcal{F}_{MajorId}(STUDENT)$
- Bir bölümü olan tüm öğrencilerin sayısını bulunuz
- Q16: $\mathcal{F}_{COUNT(MajorId)}(STUDENT)$
- Kaç farklı bölüm olduğunu bulunuz
- Q17: $\mathcal{F}_{COUNTDISTINCT(MajorId)}(STUDENT)$

STUDENT	SId	SName	GradYear	MajorId
	1	joe	2004	10
	2	amy	2004	20
	3	max	2005	10
	4	sue	2005	20
	5	bob	2003	30
	6	kim	2001	20
	7	art	2004	30
	8	pat	2001	20
	9	lee	2004	10

STUDENT(SId, SName, GradYear, MajorId)
DEPT(DId, DName)
COURSE(CId, Title, DeptId)
SECTION(SectId, CourseId, Prof, YearOffered)
ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

- En çok «A» notu alınan section'da kaç tane «A» notu alındığını bulunuz.

Q18: $\sigma_{Grade='A'}(ENROLL)$

Q19: $SectionId \mathcal{F}_{COUNT(EId)}(Q18)$

Q20: $\mathcal{F}_{MAX(CountOfEId)}(Q19)$

Q18	EId	StudentId	SectionId	Grade
	14	1	13	A
	54	4	53	A
	64	6	53	A

Q19	SectionId	CountOfEId
	13	1
	53	2

Q20	MaxOfCountOfEId
	2

ENROLL	EId	StudentId	SectionId	Grade
	14	1	13	A
	24	1	43	C
	34	2	43	B+
	44	4	33	B
	54	4	53	A
	64	6	53	A

STUDENT(SId, SName, GradYear, MajorId)
 DEPT(DId, DName)
 COURSE(CId, Title, DeptId)
 SECTION(SectId, CourseId, Prof, YearOffered)
 ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

Cartesian Product Örnek:

- Öğrencilerin ve bölümlerin tüm olası çarpımlarını bulunuz.

Q22: $STUDENT \times DEPT$

- Tüm öğrencileri ve onların okuduğu bölüm isimlerini bulunuz.

$\sigma_{MajorId=DId}(STUDENT \times DEPT)$

DEPT	DId	DName
	10	compsci
	20	math
	30	drama

SId	SName	MajorId	GradYear	DId	DName
1	joe	10	2004	10	compsci
2	amy	20	2004	10	compsci
3	max	10	2005	10	compsci
4	sue	20	2005	10	compsci
5	bob	30	2003	10	compsci
6	kim	20	2001	10	compsci
7	art	30	2004	10	compsci
8	pat	20	2001	10	compsci
9	lec	10	2004	10	compsci
1	joe	10	2004	20	math
2	amy	20	2004	20	math
3	max	10	2005	20	math
4	sue	20	2005	20	math
5	bob	30	2003	20	math
6	kim	20	2001	20	math
7	art	30	2004	20	math
8	pat	20	2001	20	math
9	lec	10	2004	20	math
1	joe	10	2004	30	drama
2	amy	20	2004	30	drama
3	max	10	2005	30	drama
4	sue	20	2005	30	drama
5	bob	30	2003	30	drama
6	kim	20	2001	30	drama
7	art	30	2004	30	drama
8	pat	20	2001	30	drama
9	lec	10	2004	30	drama

STUDENT(SId, SName, GradYear, MajorId)
 DEPT(DId, DName)
 COURSE(CId, Title, DeptId)
 SECTION(SectId, CourseId, Prof, YearOffered)
 ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

Join Örnekler:

- Tüm öğrencileri ve onların bölüm isimlerini bulunuz

Q24: $(STUDENT \bowtie_{MajorId=DId} DEPT)$

- 2004 yılında 'Joe' isimli öğrencilerin aldığı notları bulunuz

Q25: $\sigma_{SName='Joe'}(STUDENT)$

Q26: $(Q25 \bowtie_{SId=StudentId} ENROLL)$

Q27: $\sigma_{YearOffered=2004}(SECTION)$

Q28: $(Q26 \bowtie_{SectId=SectionId} Q27)$

Q29: $\Pi_{Grade}(Q28)$

STUDENT(SId, SName, GradYear, MajorId)
 DEPT(DId, DName)
 COURSE(CId, Title, DeptId)
 SECTION(SectId, CourseId, Prof, YearOffered)
 ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

- En çok «A» notunun verildiği section'ı bulunuz.
- Q30: $(Q20 \bowtie_{MaxCountOfEId=CountOfEId} Q19)$
- Q31: $(Q30 \bowtie_{SectionId=SectId} SECTION)$
- «Joe» ile aynı bölümde okuyan öğrencileri bulunuz.
- Q32: $\Pi_{MajorId} (\sigma_{SName=«Joe»} (STUDENT))$
- Q33: $\rho_{MajorId, JoesMajor} (Q32)$
- Q34: $(Q33 \bowtie_{MajorId=JoesMajor} STUDENT)$
- Q18: $\sigma_{Grade=«A»} (ENROLL)$
- Q19: $SectionId \mathcal{F}_{COUNT(EId)} (Q18)$
- Q20: $\mathcal{F}_{MAX(CountOfEId)} (Q19)$

STUDENT(SId, SName, GradYear, MajorId)
 DEPT(DId, DName)
 COURSE(CId, Title, DeptId)
 SECTION(SectId, CourseId, Prof, YearOffered)
 ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

Semi Join Örnek:

- En az 1 öğrencisi olan departmanları bulunuz
- Q35: $(DEPT \bowtie_{DId=MajorId} STUDENT)$
- Eş çözüm
- Q36: $(DEPT \bowtie_{DId=MajorId} STUDENT)$
- Q37: $DId, DName \mathcal{F} (Q36)$
- «Einstein» dan ders alan öğrencileri bulunuz
- Q38: $\sigma_{Prof=«Einstein»} (SECTION)$
- Q39: $(ENROLL \bowtie_{SectionId=SectId} Q38)$
- Q40: $(STUDENT \bowtie_{SId=StudentId} Q39)$

STUDENT(SId, SName, GradYear, MajorId)
 DEPT(DId, DName)
 COURSE(CId, Title, DeptId)
 SECTION(SectId, CourseId, Prof, YearOffered)
 ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

Anti Join Örnekler:

- Hiç öğrencisi olmayan departmanları listeleyiniz.
- Q41: $(DEPT \supset_{DId=MajorId} STUDENT)$
- Hiç 'F' notu alınmayan dersleri bulunuz.
- Q42: $(\sigma_{Grade=«F»} ENROLL)$
- Q43: $(SECTION \supset_{SectId=SectionId} Q42)$
- Üstteki soru için aşağıdaki çözüm olur mu?
- $(\sigma_{Grade \neq «F»} (ENROLL) \bowtie_{SectionId=SectId} SECTION)$
- Tüm öğrencilerinin 'A' notu aldığı dersleri bulunuz.
- Q44: $\sigma_{Grade \neq «A»} (ENROLL)$
- Q45: $(SECTION \supset_{SectId=SectionId} Q44)$

STUDENT(SId, SName, GradYear, MajorId)
 DEPT(DId, DName)
 COURSE(CId, Title, DeptId)
 SECTION(SectId, CourseId, Prof, YearOffered)
 ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

- Hiç 'F' notu vermeyen hocaları bulunuz.
- Q46: $(\sigma_{Grade \neq 'F'} ENROLL)$
- Q47: $(SECTION \bowtie_{SectId=SectionId} Q46)$
- Q48: $\rho_{Prof, BadProf}(Q47)$
- Q49: $(SECTION \supset_{Prof=BadProf} Q48)$
- Q50: $\rho_{Prof} \mathcal{F}(Q49)$
- Verdiği her dersten en az bir tane 'F' veren hocaları bulunuz.
- Q51: $\rho_{Prof, GoodProf}(Q43)$
- Q52: $(SECTION \supset_{Prof=GoodProf} Q51)$
- Q53: $\rho_{Prof} \mathcal{F}(Q52)$

Q42: $(\sigma_{Grade='F'} ENROLL)$
Q43: $(SECTION \supset_{SectId=SectionId} Q42)$

STUDENT(SId, SName, GradYear, MajorId)
DEPT(DId, DName)
COURSE(CId, Title, DeptId)
SECTION(SectId, CourseId, Prof, YearOffered)
ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

Outer Join Örnek:

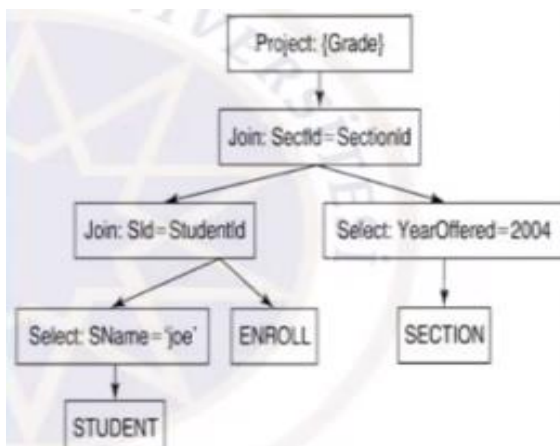
- En az 1 tane ders alan öğrencilerin numaralarını gösteriniz.
- Q54: $\rho_{StudentId} \mathcal{F}_{count}(EId)(ENROLL)$
- Tüm öğrenciler için ayrı ayrı aldığı farklı notların sayısını bulunuz.
- Q55: $(STUDENT \bowtie_{SId=StudentId} ENROLL)$
- Q56: $\rho_{SId, Grade} \mathcal{F}_{count}(EId)(Q55)$

STUDENT(SId, SName, GradYear, MajorId)
DEPT(DId, DName)
COURSE(CId, Title, DeptId)
SECTION(SectId, CourseId, Prof, YearOffered)
ENROLL(EId, StudentId, SectionId, Grade)

Öğr. Grv. Furkan ÇAKMAK

Query Tree:

Örnek: 2004'te 'joe' isimli öğrencinin aldığı notları bulan query tree:



Örnek: Einstein'dan ders alan öğrencileri bulan query tree:

